View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Universiteit Twente Repository

Journal of Educational Statistics
Fall 1989, Vol. 14, No. 3, pp. 279–290

Algorithms for Computerized Test Construction Using Classical Item Parameters

Jos J. Adema
and
Wim J. van der Linden
University of Twente

Key words: *test construction, classical test theory, item banking, linear programming*

Recently, linear programming models for test construction were developed. These models were based on the information function from item response theory. In this paper another approach is followed. Two 0–1 linear programming models for the construction of tests using classical item and test parameters are given. These models are useful, for instance, when classical test theory has to serve as an interface between an IRT-*based item banking system and a test constructor not familiar with the underlying theory.*

In this paper the construction of tests from an item bank calibrated under an item response model is considered. It is assumed that estimates of the parameters representing such properties as item difficulty, discriminating power, and the effect of random guessing are stored with the items in the bank. Given such a system, it is possible at any desired moment to construct tests with useful properties. Until now the information function was used for constructing tests (e.g., Theunissen, 1985; van der Linden & Boekkooi-Timminga, in press). In this approach tests are constructed so that a target for the information function is approximated as closely as possible. The approach has the disadvantage that the test constructor has to specify a target information function. This can be a difficult task, in particular when test constructors are unfamiliar with the underlying theory. Therefore, some test constructors may want to have the option of using classical item parameters, and the question arises as to if it is possible to use classical test theory as an interface between the item bank system and the test constructor.

Van der Linden (1986) has shown that the usual classical item and test parameters can be deduced from the item response theory (IRT) parameters under the assumption of a population distribution over the person parameter, even if the item bank is multidimensional. For the difficulty and discriminating power parameter, the reverse is also possible, provided that some conditions are met (see Lord, 1980, p. 33). It is not possible, however, to derive IRT item parameters from classical test parameters such as the

reliability coefficient, although the former are needed to calculate test information functions. Therefore, target information functions cannot be derived from classical test specifications. So, if a test constructor unfamiliar with item response theory wants to construct a test from an item bank calibrated under an IRT model, classical item and test parameters have to be used in the process of item selection. The only difference is that the parameter values need not be estimated directly from empirical data, but can be derived a priori from the IRT parameters.

Although the present research was motivated by an IRT item banking project in which the option of classical test construction was needed, it should be observed that the results in this paper do not hold only for this case. In fact, they can be used in any environment where tests are constructed using classical test and item parameters, irrespective of the number of items available and the way in which estimates of the parameters are obtained.

In the following it is assumed that, for the population of examinees considered, the item bank system has generated the classical item difficulty and discrimination parameters. Also, the test constructor not only wants a test with maximal reliability but also imposes other constraints on the tests, for instance, with respect to the range of item difficulties, the distribution of the items over subject matter, the administration time needed, the "history" of the items (e.g., the frequencies of previous usage), or the length of the test. It is obvious that in this case test construction by hand will lead to combinatorial problems if all possibilities have to be considered. It is the purpose of this paper to offer two linear programming models that can be applied to solve these problems and for which standard algorithms in computer code are amply available in textbooks (e.g., Kuester & Mize, 1973; Land & Powell, 1973; Syslo, Kowalik, & Deo, 1983), software libraries such as NAG (Numerical Algorithms Group Limited), and software packages as MPSX/370 (IBM), Lando (Center for Mathematics and Computer Science), or Lindo (Lindo Systems Inc.).

Reliability of a Test

The test construction goal we will consider is maximization of the classical reliability of the test for a given population of examinees. The reliability of a test is defined as the squared correlation between the observed and the true scores, ρ^2_{XT}. This quantity is dependent, however, on the covariance between all items (Lord & Novick, 1968, Formula 15.3.9), and it is not possible to write the reliability coefficient as a function of a more limited number of item parameters. Therefore, it is impossible to maximize the reliability of a test directly, and we shall resort to maximization of a lower bound.

A well-known and simple lower bound to test reliability is coefficient α (Lord & Novick, 1968, p. 331):

$$\alpha = n(n-1)^{-1}\left[1 - \left(\sum_{i=1}^{n} \sigma_i^2\right) \Big/ \left(\sum_{i=1}^{n} \sigma_i \rho_{iX}\right)^2\right] \tag{1}$$

where n is the number if items in the test, σ_i^2 is the variance of item i, and ρ_{iX} is the item-test correlation of item i. Other lower bounds (Guttman, 1945; Jackson & Agunwamba, 1977; Raju, 1977; ten Berge, Snijders, & Zeegers, 1981) cannot be used in our application. Because they depend on the covariances or higher order moments between the items, these lower bounds cannot be linearized satisfactorily. Also, in order to deal with covariances, decision variables have to be introduced to denote whether or not *pairs* of items should be included in the test. The number of variables required for an item bank of practical size makes the maximization of lower bounds with covariances impossible.

If no restriction were imposed on the test length, the number of items in the test would become too large, because adding items with a positive item-test covariance tends to increase the value of α. Hence we will fix the length of the test. Maximization of α then implies minimization of

$$\sum_{i=1}^{n} \sigma_i^2 \Big/ \left(\sum_{i=1}^{n} \sigma_i \rho_{iX}\right)^2 \tag{2}$$

for a fixed value of n.

Suppose the item bank consists of I items and define the decision variables x_i, $i = 1, 2, \ldots, I$ as

$$x_i = \begin{cases} 0 \text{ item } i \text{ not in the test} \\ 1 \text{ item } i \text{ in the test.} \end{cases}$$

A maximal value of α is then obtained for a solution to the following 0–1 nonlinear programming model:

$$\min \sum_{i=1}^{I} \sigma_i^2 x_i \Big/ \left(\sum_{i=1}^{I} \sigma_i \rho_{iX} x_i\right)^2 \tag{3}$$

subject to

$$\sum_{i=1}^{I} x_i = n; \tag{4}$$

$$\sum_{i=1}^{I} v_{ij} x_i \leq w_j, \qquad j = 1, 2, \ldots, J; \tag{5}$$

$$x_i \in \{0, 1\}, \qquad i = 1, 2, \ldots, I, \tag{6}$$

where constraint (4) implies that the test length equals n, and (5) has been added to deal with possible practical constraints, that is, demands that the

test constructor imposes on possible properties of the test, like the distribution of the items over subject matter, the frequency of previous usage of the items, the final date of administration of the items, and the administration time available for the test. A review of the possibilities to formulate such constraints linearly has been given elsewhere (van der Linden & Boekkooi-Timminga, in press) and will not be repeated here. Each different application of (5) will involve different definitions of v_{ij} and w_j.

There is no efficient algorithm for solving integer nonlinear programming models (Rao, 1985), so the above model cannot be used in practice. If the problem can be formulated as a 0–1 linear programming model, however, more efficient algorithms are available.

Formulation of the Problem as a 0–1 LP Model

In this section, two 0–1 linear programming models (0–1 LP models) will be formulated. The performance of these models will be compared in a simulation study. The possibility to use expressions like (5) to include practical constraints in the test construction process is skipped during the presentation of the models, but will be taken up again in an example at the end of the paper.

Inspection of (3) shows that both of its sums are linear in the decision variables. This suggests an approach in which one of these expressions is used as an objective function and the other as a constraint. Because for a wide range of possible values of π, the classical difficulty of an item, the numerator of (3) varies less than the denominator, α can be expected to depend more strongly on the latter. This effect is verified empirically in Ebel (1967). Also, if the numerator of (3) is restricted to a low (high) value, the denominator takes on a low (high) value, because σ_i figures both in the numerator and in the denominator. So a restriction on the numerator probably does not influence the value of α very much. Therefore, it seems sensible to maximize the denominator of (3), constraining the numerator to a low value. This is realized in the following model.

Test Construction Model I

$$\max \sum_{i=1}^{I} \sigma_i \rho_{iX} x_i \tag{7}$$

subject to

$$\sum_{i=1}^{I} \sigma_i^2 x_i \leq c; \tag{8}$$

$$\sum_{i=1}^{I} x_i = n; \tag{9}$$

$$\sum_{i=1}^{I} v_{ij} x_i \leq w_j, \qquad j = 1, 2, \ldots, J; \tag{10}$$

$$x_i \in \{0, 1\}, \qquad i = 1, 2, \ldots, I, \tag{11}$$

where $c > 0$ is a constant. This model is linear in its variables and can be solved for $(x_1, \ldots, x_I)$ by a branch-and-bound method (Dakin, 1965; Land & Doig, 1960).

The choice of a value for c can be motivated as follows. The maximal possible value of the sum of the item variances in the model is equal to $n/4$. In addition, the numerator and denominator of (2) have σ_i as a common factor. Therefore, if c approaches its maximum, a maximal value will be found, but at the same time the numerator will tend to be too large. On the other hand, if c approaches its minimum, a minimal value for the numerator will be attained, at the cost of a constrained denominator. The latter is due not only to the common factor σ_i, but also to a restriction-of-range effect on ρ_{iX}. Hence, the optimal value of c will tend to be closer to $n/4$ than to zero. This issue will be pursued further in the section on empirical results below.

Because the variances of the items are not as important as the item discriminations, the following 0–1 LP model, which selects items with the highest discriminations, is an alternative to Test Construction Model I.

Test Construction Model II

$$\max \sum_{i=1}^{I} \rho_{iX} x_i \tag{12}$$

subject to

$$\sum_{i=1}^{I} x_i = n; \tag{13}$$

$$\sum_{i=1}^{I} v_{ij} x_i \leq w_j, \qquad j = 1, 2, \ldots, J; \tag{14}$$

$$x_i \in \{0, 1\}, \qquad i = 1, 2, \ldots, I. \tag{15}$$

The advantage of this model is that we do not have to choose a value for c. The approach implemented in Model II is recommended in Gulliksen (1950, p. 379). The application of complex engineering methods like mathematical programming methods is useful because in practice constraints like (14) are involved (see the example below). Otherwise, selection by hand would be preferable.

Empirical Validation and Comparison

In this section the assumptions underlying model (7)–(11) are verified empirically, and the performances of the models in (7)–(11) and (12)–(15) are compared. An example at the end of this section illustrates the possibility of including practical constraints in the model.

Two item banks of 500 items were generated. For Item Bank 1, the underlying response model was the Rasch model with item parameters drawn from the distribution $N(-0.5, 1)$. For Item Bank 2, the underlying model was the 3-parameter logistic model with item parameters a_i and b_i drawn from the distributions $U(0.5, 1.5)$ and $U(-3, 3)$, respectively. The guessing parameters c_i were set equal to 0.2 for the first 250 items and equal to 0.0 for the other items. To estimate the classical item difficulties, π_i, and item discriminations (i.e., item-test correlations where the whole item bank was considered as a test), ρ_{iB}, 1,000 examinees ($\theta \sim N(0, 1)$) were generated to answer the items.

The computer program Lando was used to solve the 0–1 linear programming models on a DEC2060 computer. Because it takes too much time to find a 0–1 solution for the model in (7)–(9) and (11) directly, the relaxation of this model was solved, that is, the model with constraints $0 \leq x_i \leq 1$ instead of $x_i \in \{0, 1\}$ for $i = 1, 2, \ldots, I$. This could be done, because it is known that the number of fractional values in the solution is not greater than the number of constraints (Dantzig, 1957). Therefore, the solution to the model in (7)–(9) and (11) was found by rounding fractional values for at most two of the I decision variables.

The model assumptions were first verified by comparing tests from Item Bank 1 for different values of c. The number of items in the tests was 20. Table 1 shows the values of coefficient α. In this table, α^* denotes coefficient alpha with ρ_{iB} replacing item ρ_{iX}, whereas α is the exact value of the coefficient calculated after the test was selected.

Table 1 shows that the differences between values of α of tests constructed for different values of c were small. However, high values of c generally gave the best results.

For Item Bank 2 (3-parameter model), again tests were constructed for

TABLE 1
Coefficient α for tests constructed from a simulated Rasch calibrated item bank (n = 20) using model (7)–(9), (11)

c	α^*	α
5.0	.8096	.8478
4.5	.8028	.8413
4.0	.7803	.8252
3.5	.7491	.8069

Note. α^* and α are the coefficients α based on ρ_{iB} and ρ_{iX}, respectively. Parameter c is given in constraint (8).

TABLE 2
Coefficient α for tests constructed from a simulated item bank calibrated under the 3-parameter logistic model (n = 20) using model (7)–(9), (11)

c	α^*	α
5.0	.8395	.8712
4.5	.8388	.8678
4.0	.8288	.8559
3.5	.8008	.8401
3.0	.7696	.8205

Note. α^* and α are the coefficients α based on ρ_{iB} and ρ_{iX}, respectively. Parameter c is given in constraint (8).

TABLE 3
Coefficient α for tests constructed from both simulated item banks using models (7), (9), (11), and (12)–(13), (15)

		Model (7), (9), (11)		Model (12)–(13), (15)	
Item bank	n	α^*	α	α^*	α
1	20	.8096	.8478	.8107	.8465
	40	.9013	.9122	.9020	.9122
2	20	.8395	.8712	.8411	.8723
	40	.9088	.9189	.9114	.9210

Note. α^* and α are the coefficients α based on ρ_{iB} and ρ_{iX}, respectively.

different values of c. The results are displayed in Table 2. Once more, the best results tended to be found for high values of c. Therefore, it seems possible to choose c maximal, implying that constraint (8) becomes redundant and can be omitted.

In Table 3, a comparison is made between model (7), (9), (11) (with c maximal), and model (12)–(13), (15). The numbers of items in the tests were equal to 20 or 40, and the models were applied to both item banks. Table 3 demonstrates that the model in (12)–(13) and (15) gave excellent results. The values of α were as good as for the best choices of c in Tables 1 and 2.

Practical Constraints in Test Construction

The models considered so far in the simulation study are not realistic, because practical constraints were not included. In fact, it is the presence

of such constraints that forces us to use algorithms for solving the combinatorial problems involved. Therefore, the two models were extended with the following illustrative constraints:

$$\sum_{i=1}^{500} t_i x_i \leq 35*n; \tag{16}$$

$$\sum_{i=1}^{250} x_i \geq 0.5*n; \tag{17}$$

$$\sum_{i=251}^{500} x_i \geq 0.25*n; \tag{18}$$

$$\sum_{i=1}^{125} x_i + \sum_{i=251}^{375} x_i \geq 0.4*n; \tag{19}$$

$$\sum_{i=126}^{250} x_i + \sum_{i=376}^{500} x_i \geq 0.4*n; \tag{20}$$

$$0.45*n \leq \sum_{i=1}^{500} p_i x_i \leq 0.55*n; \tag{21}$$

$$x_i \leq e_i*10; \qquad i = 1, \ldots, 500. \tag{22}$$

Constraint (16) implied that the administration time of the test was not allowed to exceed $35*n$ seconds. The parameter t_i in this inequality was the estimated administration time for item i, for example, an estimate of the 95th percentile of the distribution of time needed to solve item i in the population of examinees. The item banks were supposed to be divided into two subsets. The first subset consisted of 250 multiple-choice items; the other of 250 essay items. The constraint in (17) stipulated that at least half of the items in the test were multiple-choice items, whereas (18) guaranteed that at least a quarter of the items in the test were essay items. The item banks were also divided into subsets by content. The items with index values equal to 1–125 and 251–375 were grammar items. The other items were vocabulary items. Thus, constraint (19) and (20) implied that at least 40% of the items in the test were grammar and vocabulary items, respectively. The coefficients p_i, $i = 1, \ldots, 500$, in constraint (21) were the estimated classical item difficulties. According to constraint (21) the mean item difficulty of the test should be in the range .45–.55. Suppose the test has to be administered to a population of men and women and that for this application some of the items are biased. For each item the hypothesis has been tested that men and women have identical response functions. Let e_i be the one-sided probability of exceedance under the null hypothesis of no bias. The hypothesis has to be rejected for probabilities smaller than .10. Constraints (22) select only test items for which the hypothesis of no bias holds.

The extended models were not solved by rounding the optimal solution to the relaxed model, because this could yield an infeasible solution. A heuristic based on modifications in the branch-and-bound method was used. Papadimitriou and Steiglitz call the branch-and-bound method an approach in which

> we try to construct a proof that a solution is optimal, based on successive partitioning of the solution space. The branch in branch-and-bound refers to this partitioning process; the bound refers to upper bounds that are used to construct a proof of optimality without exhaustive search. (1982, p. 433)

In our case the solution space was partitioned successively by setting variables at their lower and upper bounds (0 and 1). We started with the solution space formed by the constraints (13), (16)–(22) and $0 \leq x_i \leq 1$, $i = 1, \ldots, 500$. The upper bound z_{LP} to the objective function value of the best 0–1 solution was computed by solving the LP problem (12), (13), (16)–(22), or (7), (9), (16)–(22) and $0 \leq x_i \leq 1$ with the simplex method (see, e.g., Papadimitriou & Steiglitz). Solution spaces with a corresponding LP problem for which no feasible solution existed, or for which the objective function value (upper bound) was smaller than the objective function value z^* for the best 0–1 solution found so far, were not partitioned any further, because they could not contain the best 0–1 solution. The modifications in the heuristic were as follows.

1. A large number of variables were fixed after solving the first relaxed problem using the reduced costs (see, e.g., Murtagh, 1981, p. 25).
2. z^* was not initialized by $z^* = -\infty$ as usual but by $z^* = KZ_{LP}$, where K is a constant close to 1 ($0 \ll K < 1$); also, the first 0–1 solution found during the search process was accepted.

Both modifications are based on the small difference between the objective function values for the solution of the relaxed 0–1 and the 0–1 LP problem in test construction problems.

The latter modification was such that the size of the possible error was under control, because the maximum possible difference between the values of the objective function for the optimal solution to the relaxed model and for the 0–1 solution could be set in advance by choosing a value for K. It should be observed that such solutions always meet the constraints in the model. More details about the heuristic are given in Adema (1988).

Table 4 is similar to Table 3; Table 4, however, presents results for the extended models. The maximal possible error was chosen to be 1% of the objective function value for the optimal solution to the relaxed model. In Table 4, the results for the model with objective function (12) were again slightly better.

Finally, Tables 1–4 show that it is possible to construct tests with ρ_{iX} replaced by ρ_{iB}, because generally tests with a high value for α^* also have a high value for α.

TABLE 4
Coefficient α for tests constructed from both simulated item banks using models (7), (9), (11), (16)–(22) and (12), (13), (15)–(22)

		Model (7), (9), (11), (16)–(22)		Model (12), (13), (15)–(22)	
Item bank	n	α^*	α	α^*	α
1	20	.7966	.8393	.8023	.8413
	40	.8971	.9080	.8970	.9081
2	20	.8017	.8437	.8083	.8456
	40	.8887	.9024	.8913	.9042

Note. α^* and α are the coefficients α based on ρ_{iB} and ρ_{iX}, respectively.

Discussion

Two 0–1 linear programming models were proposed for the construction of tests using classical item parameters. Simulations were conducted to verify the assumptions underlying model (7)–(11). Ample experience with the model for various types of data (Adema, 1987) has shown that the solution invariably produces the maximal value for α for c close to $n/4$ (maximum of $\Sigma\sigma_i^2$ in the model). For example, for $I = 500$, all simulations produced the maximum of alpha for c in the neighborhood of 95% of $n/4$. Also, the optimal value of α increased monotonically with c to the point at which the maximum was obtained and then showed a monotonic but slight decrease. Therefore, for large item banks, $I > 500$, say, it is recommended to set c at its maximal value. Model (12)–(15) produced results comparable to those for model (7)–(11), in most cases producing results even slightly better. If no practical constraints have to be met, the models in (7), (9), (11) and (12)–(13), (15) can be solved by a simple algorithm that picks the items with the largest values for ρ_{iB} and $\sigma_i\rho_{iB}$, respectively. In practical situations, however, constraints on the contents of the test are always available, and then a formulation of the problem as a 0–1 LP model is needed. Such models can always be solved by the heuristic used in this paper (Adema, 1988).

References

Adema, J. J. (1987). *Toetsconstructie met klassieke item- en test parameters* [Test construction using classical item and test parameters] (Rapport 87-1). Enschede, The Netherlands: University of Twente, Department of Education.

Adema, J. J. (1988). *A note on solving large-scale zero–one programming problems* (Research Report 88-4). Enschede, The Netherlands: University of Twente, Department of Education.

Dakin, R. J. (1965). A tree-search algorithm for mixed integer programming problems. *Computer Journal, 8,* 250–255.

Dantzig, G. (1957). Discrete-variable extremum problems. *Operations Research, 5,* 266–277.

Ebel, R. L. (1967). The relation of item discrimination to test reliability. *Journal of Education Measurement, 4,* 125–128.

Gulliksen, H. (1950). *Theory of mental tests.* New York: Wiley.

Guttman, L. A. (1945). A basis for analyzing test-retest reliability. *Psychometrika, 10,* 255–282.

Jackson, P. H., & Agunwamba, C. C. (1977). Lower bounds for the reliability of the total score on a test composed of non-homogeneous items: 1. Algebraic lower bounds. *Psychometrika, 42,* 567–578.

Kuester, J. L., & Mize, J. H. (1973). *Optimization techniques with Fortran.* New York: McGraw-Hill.

Land, A. H., & Doig, A. G. (1960). An automated method for solving discrete programming problems. *Econometrica, 28,* 497–520.

Land, A. H., & Powell, S. (1973). *Fortran codes for mathematical programming: Linear, quadratic and discrete.* London: Wiley.

Lord, F. M. (1980). *Applications of items response theory to practical testing problems.* Hillsdale, NJ: Erlbaum.

Lord, F. M., & Novick, M. R. (1968). *Statistical theories of mental test scores.* Reading MA: Addison-Wesley.

Murtagh, B. A. (1981). *Advanced linear programming: Computation and practice.* New York: McGraw-Hill.

Papadimitriou, C. H., & Steiglitz, K. (1982). *Combinatorial optimization: Algorithms and complexity.* Englewood Cliffs, NJ: Prentice-Hall.

Rao, S. S. (1985). *Optimization: Theory and applications.* New Delhi: Wiley Eastern.

Raju, N. S. (1977). A generalization of coefficient alpha. *Psychometrika, 42,* 549–565.

Syslo, M. M., Deo, N., & Kowalik, J. S. (1983). *Discrete optimization algorithms: With Pascal programs.* Englewood Cliffs, NJ: Prentice-Hall.

ten Berge, J. M. F., Snijders, T. A. B., & Zeegers, F. E. (1981). To the reliability and constrained minimum trace factor analysis. *Psychometrika, 46,* 201–213.

Theunissen, T. J. J. M. (1985). Binary programming and test design. *Psychometrika, 50,* 411–420.

van der Linden, W. J. (1986) Item banking met een dialoog gebaseerd op klassieke item- en testparameters [Item banking with a dialogue based on classical item and test parameters]. In G. R. Buning, T. J. H. M. Eggen, H. Kelderman, & W. J. van der Linden (Eds.), *Het gebruik van het Raschmodel voor een decentraal toetsservicesysteem* (Rapport 86-3; pp. 1–25). Enschede, The Netherlands: University of Twente, Department of Education.

van der Linden, W. J., & Boekkooi-Timminga, E. (in press). A maximin model for test design with practical constraints. *Psychometrika.*

Authors

JOS J. ADEMA, Research Associate, University of Twente, Department of Education, P.O. Box 217, 7500 AE Enschede, The Netherlands. *Specialization:* operations research.

WIM J. VAN DER LINDEN, Professor, University of Twente, Department of Education, P.O. Box 217, 7500 AE Enschede, The Netherlands. *Specializations:* psychometric theory, data analysis, research methodology.